COLUMBIA UNIVERSITY

SYSTEMS RESEARCH GROUP

DEPARTMENT OF
ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING AND
APPLIED SCIENCE
NEW YORK, N.Y. 10027

CR 134559

Technical Report No. 107

Computer-aided Analysis of Nonlinear Networks

by

Thomas E. Stern

July, 1968

Vincent R. Lalli
Project Manager
Lewis Research Center
Office of Reliability
& Quality Assurance
21000 Brookpark Rd
Cleveland, OH 44135

This work was partially supported by the National Science Foundation under NSF GP-2789, GP-14514, GK 2283, the Office of Naval Research under Contract No. NONR 4259(04), and the National Aeronautics and Space Administration under Contract No. NGR 33-008-090.

COMPUTER-AIDED ANALYSIS OF NONLINEAR NETWORKS

by

Thomas E. Stern
Department of Electrical Engineering
Columbia University

1. INTRODUCTION

As the importance of digital electronics continues to grow so too does the need for computer-aided analysis of the highly nonlinear networks that are typical of current pulse and digital circuit technology.

In principle the time domain analysis of nonlinear networks is simply a matter of solving a set of simultaneous nonlinear differential equations; presumably a routine task for a digital computer. As we shall show, however, there are certain characteristics of the types of networks mentioned above which pose some special difficulties not normally encountered in other areas of application. Some of these difficulties manifest themselves in the formulation of the equations of the networks and others appear in integrating these equations. Furthermore, when one attempts to circumvent difficulties at one stage, they often reappear later in a different guise.

The purpose of this paper is to present a computational technique which avoids certain problems commonly encountered in existing methods. (Much of the theoretical basis for this technique has been described in an earlier paper. [1]) In Section 2 we illustrate the general characteristics of a computer-based approach to nonlinear network analysis by means of a simple example. Section 3 is devoted to a brief critique of some representative approaches to the problem. In Section 4 a new computational method based on "augmentation" is described. Some experiments comparing this method with a widely used existing computer program are described in Section. 5. In Section 6 some of the open questions concerning the method of augmentation are discussed.

2. CHARACTERISTICS OF THE COMPUTER-BASED APPROACH

It is helpful to consider the network analysis problem in three phases:

1) Modelling
2) Equation formulation
3) Numerical solution

As we shall see, the choice of alternatives at each step has a direct effect on the implementation of subsequent steps. Because of this, the network analysis problem must be considered in its entirety without isolating one step from another.

The astable transistor multivibrator shown in Fig. 1 will be used as an illustrative example throughout the paper. Clearly the first step in the analysis must be to choose an appropriate model for the transistors. Two possible alternatives are shown in Fig. 2a, b. The first, a low frequency model containing no reactive elements would seem to be appropriate for examining the gross features of circuit behavior — essentially all features except the detailed shape of the leading and trailing edges of the waveforms. For a closer approximation of the high-frequency effects (e.g. risetime) the second high-frequency model would be more suitable. Let us observe the consequences of each of these choices in implementing step (2), equation formulation.

The state variable formulation of the network equations is the obvious choice in a computer-based analysis. Several more or less equivalent methods of deriving these equations are available. [2, 3, 4] Considering first, the low frequency transistor model and choosing the charges x_1 and x_2 on capacitors C_1 and C_2 respectively as state variables, we obtain the following equations

$$\overset{\circ}{x}_1 = \frac{1}{R}\left(u_3 - u_5 - \frac{x_1}{C_1} + E\right) + \frac{1}{R_B}\left(u_3 - u_5 - \frac{x_1}{C_1} + u_6\right) \triangleq f_1(x, u) \qquad (1a)$$

$$\overset{\circ}{x}_2 = \frac{1}{R}\left(u_4 - u_6 - \frac{x_2}{C_2} + E\right) + \frac{1}{R_B}\left(u_4 - u_6 - \frac{x_2}{C_2} + u_5\right) \triangleq f_2(x, u) \qquad (1b)$$

$$0 = - f_1(x, u) - F_C(u_3, u_5) - \frac{1}{R_L}\left(u_3 - u_5 + E\right) \triangleq g_3(x, u) \qquad (1c)$$

$$0 = - f_2(x, u) - F_C(u_4, u_6) - \frac{1}{R_L}\left(u_4 - u_6 + E\right) \triangleq g_4(x, u) \qquad (1d)$$

$$0 = f_1(x, u) - F_E(u_3, u_5) + \frac{1}{R_B}\left(\frac{x_2}{C_2} - u_4 - u_5 + u_6\right) + \frac{1}{R_L}\left(u_3 - u_5 + E\right)$$

$$\triangleq g_5(x, u) \qquad (1e)$$

$$0 = f_2(x, u) - F_E(u \quad u_6) + \frac{1}{R_B}\left(\frac{x_1}{C_1} - u_3 - u_6 + u_5\right) + \frac{1}{R_L}\left(u_4 - u_6 + E\right)$$

$$\triangleq g_6(x, u) \qquad (1f)$$

where the variables u_3, u_4, u_5, u_6 represent voltages within the transistor models. (See Fig. 3.) Note that these equations are not in state variable form, but rather in the form of differential equations plus constraints, containing certain auxiliary variables $u = (u_3 \cdots u_6)$ as well as the state variables $x = (x_1, x_2)$.

This is a fundamental characteristic of the equations of any lumped network, and thus a canonical form for the network equations may be taken as

$$\dot{x} = f(t, x, u) \tag{2a}$$

$$0 = g(t, x, u) \tag{2b}$$

where x represents a set of n state variables, u a set of k auxiliary variables, (2a) a set of n first order differential equations and (2b) a set of k constraint equations.

If we were to use the high frequency transistor model (introducing the additional capacitances shown dotted in Fig. 3) Eq. (1) would be modified to the form

$$\dot{x}_1 = f_1(x, u) \tag{3a}$$

$$\dot{x}_2 = f_2(x, u) \tag{3b}$$

$$\dot{u}_3 = \frac{1}{C_C(u_3)} g_3(x, u) \tag{3c}$$

$$\dot{u}_4 = \frac{1}{C_C(u_4)} g_4(x, u) \tag{3d}$$

$$\dot{u}_5 = \frac{1}{C_E(u_5)} g_5(x, u) \tag{3e}$$

$$\dot{u}_6 = \frac{1}{C_E(u_6)} g_6(x, u) \tag{3f}$$

Note that the only difference between Eq. (2) based on the low frequency model and Eq. (3) based on the high frequency model is that the auxiliary variables have now become state variables and the constraint relations have now become differential equations.

The consequences of the two alternative choices of transistor model become apparent in attempting to solve the network equations. Consider first, the general constrained form (2) which results from the low frequency model. Since these equations cannot be integrated by standard numerical methods, an obvious approach would be to attempt to eliminate the constraints by solving (2b) for

$$u = h(t, x) \tag{4}$$

and substituting (4) into (2a) to give

$$\dot{x} = f[t, x, h(t, x)]$$

which is in the normal state variable form

$$\dot{x} = F(t, x) \tag{5}$$

If this operation can be performed uniquely (that is, if a unique solution (4) of the constraint equations exists) and if the resultant differential equations possess unique solutions defined in the future for all initial states, we shall say that the network equations are <u>determinate</u>. Otherwise they shall be called indeterminate. The question of determinacy of the network equations is central to any analysis problem, computer-based or otherwise, for unless the equations are determinate it is doubtful whether they have any meaningful physical interpretation. The crux of the problem of determinacy lies in the form of the constraint equations (2b). Regarding these we must ask:

a) Does a unique solution of the form (4) exist?

b) If so, how can it be computed?

c) If not, how can the network be analyzed?

Returning to our example, a little thought will indicate that it is impossible to know whether a unique solution of Eq. (1c-f) exists (much less how to compute it) without knowing the numerical values of the various parameters involved. Moreover, it is well-known that the combinations of parameters which lead to proper multivibrator action are just those which lead to multiple solutions of the constraint equations. The low frequency transistor model in this circuit therefore necessarily leads to indeterminate equations. (However, it is well to recall that this is the model generally employed in most non-computer-based analyses of the multivibrator.)

On the other hand, choosing the high frequency transistor model we obtain an unconstrained system, albeit one of higher order, and thus the question of determinacy does not arise[1]. It might therefore appear that it is always desirable to choose a network model in which the problem of constraints does not occur. This, in fact, is one fairly common approach used in computer-based analysis. (See Section 3.) Let us explore the consequences of this choice a bit further. Considering realistic values of the twansistor capacitances, it would not be unusual for the time constants associated with these capacitances to be of the order of nanoseconds while the period of the multivibrator might be of the order of milliseconds. Thus, the time period over which a solution is to be computed may be of the order of 10^6 times the shortest time constant in the system. Systems of differential equations having this property have been termed "stiff" systems. [5] Now, to avoid numerical instability, the stepsize in numerical integration schemes must generally be kept considerably smaller than the shortest time constant. Therefore, stiff systems generally demand extremely large numbers of integration steps, and can over-tax the capabilities of the fastest computers. Because conventional numerical integration routines cannot cope

[1] A system of the form (5) can also be indeterminate if its right-hand side does not possess certain smoothness and boundedness properties. However, this type of indeterminacy normally does not occur in models of physical systems.

with stiff systems in an efficient manner, and because such systems are characteristic of nonlinear electronic problems, considerable effort has recently been devoted to this subject. [6-10] Although progress has been made in developing algorithms especially suited to these systems, the problem remains difficult.

To recapitulate, we have illustrated by means of the example of the multivibrator, that in the case of highly nonlinear networks, depending upon the type of model chosen, one generally obtains a formulation either in terms of constrained differential equations, or stiff systems of differential equations. Each of these alternatives presents special problems at the final stage of the analysis.

3. CRITIQUE OF EXISTING NETWORK ANALYSIS PROGRAMS

A large variety of computer programs for network analysis are currently available, many of which are in rather widespread use. The way in which the problem of constraints is treated in each of these programs determines to a large extent its basic structure. Let us therefore digress for a moment to delineate two important classes of networks in which unique solutions of the constraint equations can be found.

1) Linearly reducible networks: When certain topological restrictions are fulfilled it can be shown that all constraints can be eliminated by linear operations alone (i.e., direct substitution and matrix manipulation). (Using the low frequency transistor model, the multivibrator equations are not linearly reducible, but using the high frequency model they are.)

2) Iteratively reducible networks: In some classes of networks a unique solution of the constraint equations (2b) can be

shown to exist although it cannot necessarily be expressed in closed form. Among these networks one can determine classes in which a particular iterative computational algorithm will always converge to the solution of (2b).

It is worth noting, however, that only rather special network configurations fit into the above two classes.

We shall now classify a representative sample of existing network analysis programs with respect to their handling of the constraint problem.

Type A) Restricted to linearly reducible networks. Certain computer programs restrict the admissible models of nonlinear network elements in such a way that the resultant networks will necessarily be linearly reducible. [11, 12] (For example, these programs would not accept our low frequency transistor model.)

Type B) Restricted to iteratively reducible networks. Using somewhat less stringent restrictions than employed in (A) above, some programs (for example [13]) will analyze successfully all networks amenable to a certain computational algorithm.

Type C) Unrestricted. Some programs allow the user to define network elements in a very general way by means of appropriate subroutines. [14, 15] As a result the constraint equations (2b) are in effect hidden within the subroutines. Thus, there is no recognition by the program that these constraints even exist.

Type D) Fixed algorithm, unrestricted networks. Here, some standard iterative procedure, for example the Newton-Raphson method, is used to solve the constraint equations. Few restrictions are placed on the form of these equations.

Each of the above types of programs has obvious drawbacks. Although types A and B have the advantage of guaranteeing that the constraint equations will be solved in a reliable manner, they seriously limit the generality of the admissible networks. Furthermore, the topological restrictions imposed often force the user to include in the description of a network various small parameters which he may prefer to neglect. (The transistor capacitances in the multivibrator circuit are a case in point.) As was pointed out earlier, the inclusion of small parameters generally leads to the problem of stiff systems of differential equations.

The limitations of a program of type C are the most serious. Since constraints are not properly recognized in the computation, it is possible to obtain completely erroneous results. A simple example illustrates this point. A program of type C was used to analyze the circuit of Fig. 4. (Note that the current-controlled voltage source, $V = R_2I$ is equivalent to a resistor.) In response to appropriate input data the program formulated various expressions describing the network, among which was the statement:

$$I = \frac{1}{R_1}\left[E - R_2 I\right]$$

If this linear constraint relation were treated by the computer as an equation it could have been solved for the auxiliary variable I. However, it was treated instead as an arithmetic assignment statement; that is, each time a new value of I was called for, the operation

$$I(NEW) = \frac{1}{R_1}\left[E - R_2 I(OLD)\right]$$

was executed. Thus, in effect, the program was performing one iteration on the constraint equation each time a solution of this equation was called for. This led to completely false results since even a repeated iteration will only converge when $|R_2/R_1| < 1$.

Programs of type (D) can be considered a compromise between the restricted typeś (A) and (B) and the unrestricted type (C). Since few restrictions are placed on the constraint equations, difficulties may be encountered if they do not possess unique solutions or if the iteration fails to converge. However, the likelihood of erroneous results in type (D) programs is considerably smaller than in type (C) programs.

4. THE METHOD OF AUGMENTATION

We present in this section a method of analysis which avoids the twin problems of eliminating nonlinear constraints and of solving stiff systems of differential equations. It is based on augmentation of the network model with elements of infinitesimally small values. The discussion will be divided into three parts:

Formulation of the network equations

Augmentation of the network model

Numerical solution of the augmented equations

4.1 Equation Formulation. Let the network be described by a connected graph G in which each branch is classified as either independent voltage source (V), capacitive (C), resistive (R), inductive (L) or independent current source (J). It will be assumed that:

1) The terminal characteristics of all capacitive branches are given in the "charge-controlled" form:

$$e_C = f_C(t, q_C) \qquad \dot{q}_C = i_C \tag{5}$$

where e_C, q_C, i_C are vectors representing respectively the capacitor voltages, charges and currents. Any capacitor whose terminal relation can be written in the form

$$e_{C_i} = S_i(t)\, q_{C_i} \qquad \text{where } S_i(t) > 0 \quad \forall\, i$$

will be called a linear positive capacitor (LPC). Coupling is permitted among all capacitive branches except LPC's.

2) The terminal characteristics of all inductive branches are given in the "flux-controlled" form:

$$i_L = f_L(t, \lambda_L) \qquad \dot{\lambda}_L = e_L \tag{7}$$

where i_L, λ_L, and e_L represent respectively the inductor currents, flux-linkages and voltages. Linear positive inductors (LPL) are defined in the same way as LPC's. Coupling is permitted among all inductive branches except LPL's.

3) The terminal characteristics of the resistive branches are given in the form

$$w = f_R(t, y) \tag{8}$$

where each element w_i of the vector w and each element y_i of y is a resistive branch voltage or current, and y_i is a current (voltage) whenever w_i is a voltage (current). We partition the resistive branches into three disjoint classes:

a) Linear positive resistance (LPR), defined as is the LPC.

b) Voltage controlled nonlinear resistance (VCNR) whenever the branch is not LPR and w_i is a current.

c) Current controlled nonlinear resistance (CCNR) whenever the branch is not LPR and w_i is a voltage.

We shall call w the controlled variables and y the controlling variables. (All elements which are not LPC, LPL or LPR will be termed nonlinear even though their terminal relations may be linear.) Coupling is permitted among all resistive branches except LPR's. (Note that the classifications VCNR and CCNR are based on the way the terminal relations are given, even though it may be possible to express them in other forms.)

4) There are no loops consisting of voltage sources only or of voltage sources and capacitors only, nor cutsets consisting of current sources only or current sources and inductors.

(Certain of these assumptions were made merely to simplify the exposition. For example, assumption (4) could be partially relaxed without any radical change in the results which follow.)

The network equations are obtained by appropriately combining the terminal relations (6,7,8) with the Kirchhoff law constraints defined by the graph G. We begin by selecting a "proper tree", defined to be one whose elements are chosen in the following order of priority: LPC's, nonlinear capacitors, independent voltage sources, VCNR's, LPR's, CCNR's, independent current sources, nonlinear inductors, LPL's. Following [1] the state variables are chosen as cutset charges q and loop flux linkages λ. When there are no all-capacitor loops and no all inductor cutsets, each capacitor charge and inductor flux-lindage constitutes a state variable. Otherwise the state variables are linear combinations of these charges and flux linkages. (See Appendix.)

With respect to any proper tree T, we define certain "excess" elements[1], associating with each excess element an "auxiliary

[1] Our usage of the term "excess" is somewhat different from that commonly appearing in the literature.

variable" as follows:

Excess elements	Auxiliary variables
All nonlinear capacitive chords of T	charges q_c
All nonlinear inductive branches of T	flux linkages λ_ℓ
All VCNR branches of T	voltages e_y
All CCNR chords of T	currents i_z

Note that for an element to be excess it must be nonlinear and it must appear in a certain topological configuration. (For example, a nonlinear capacitor would be excess if and only if it occurred as a chord in an all-capacitor loop.)

It is now possible to reduce all of the network data by linear operations only to the form

$$\dot{q} = f_q(t, q, \lambda, q_c, \lambda_\ell, e_y, i_z) \tag{9a}$$

$$\dot{\lambda} = f_\lambda(t, q, \lambda, q_c, \lambda_\ell, e_y, i_z) \tag{9b}$$

$$0 = g_c(t, q, \lambda, q_c, \lambda_\ell, e_y, i_z) \tag{9c}$$

$$0 = g_\ell(\qquad \text{"} \qquad) \tag{9d}$$

$$0 = g_y(\qquad \text{"} \qquad) \tag{9e}$$

$$0 = g_z(\qquad \text{"} \qquad) \tag{9f}$$

Note that Eqs. (9) are in the general canonical form (2). The state variables are $x = (q, \lambda)$, the auxiliary variables, $u = (q_c, \lambda_\ell, e_y, i_z)$, and there are exactly as many constraint equations as there are auxiliary variables. In fact, there is a one-one correspondence between constraint equations and excess elements. A derivation and explicit representation of the functions f and g in (9) is presented in the Appendix. It is worth noting that the selection of the proper tree and formulation of these functions is a straightforward procedure for a digital computer.

As an example, for the multivibrator with low frequency transistor model, Eqs. (9) take the form (1) where $x = q = (x_1, x_2)$, the charges on the capacitors C_1 and C_2, and $u = e_y = (u_3, u_4, u_5, u_6)$, the voltages across the excess VCNR's in the transistor models.

4.2 Augmentation:

As was pointed out previously, the constrained equations (9) are not in a form suitable for numerical integration. Furthermore, in the general case, there will be no systematic way of eliminating the constraints and the auxiliary variables. Therefore, to avoid the problem of constraints we shall augment the original network by inserting a number of "stray" network elements of infinitesimally small values. The augmentation will be performed in such a way as to convert each implicit constraint equation to a differential equation. The procedure is illustrated in Fig. 5. We insert exactly as many augmenting elements as there are "excess" elements in the original network. Each of the four classes of excess elements requires a different type of augmentation. The case of the excess VCNR's will be taken as an illustrative example. As shown in Fig. 5, each excess VCNR is augmented by a stray capacitance of arbitrarily small positive value ϵ. Before augmentation the terminal characteristic of the element is of the form $i_y = f_y(\cdots e_y \cdots)$, that is, the current i_y through the element may depend on its own voltage e_y as well as various other variables. After addition of the capacitance ϵ the terminal relation becomes

$$i_y = \epsilon \overset{\circ}{e}_y + f_y(\cdots e_y \cdots)$$

Thus, if we now let i_y and e_y be vectors representing respectively the currents and voltages on all excess VCNR's then the effect of the augmentation is to add a term of the form $\epsilon \dot{e}_y$ to all equations in which the currents i_y appear. It can be seen from Eq. (A-9e) that these currents appear in (and only in) Eq. (9e), so that the net result of this augmentation is to add the term $\epsilon \dot{e}_y$ to the left-hand side of Eq. (9e) converting it to a differential equation. Similar reasoning follows for the other three types of augmentation illustrated in Fig. 5. The end result is that the constraint equations (9c-f) are now converted to the following differential form.[1]

$$\epsilon \dot{q}_c = g_c(t, x, u) \qquad (10a)$$

$$\epsilon \dot{\lambda}_\ell = g_\ell(t, x, u) \qquad (10b)$$

$$\epsilon \dot{e}_y = g_y(t, x, u) \qquad (10c)$$

$$\epsilon \dot{i}_z = g_z(t, x, u) \qquad (10d)$$

Note that the order of the augmented system is equal to the order of the unaugmented system plus the number of excess elements. The augmentation with small stray elements described in Fig. 5 can be viewed as a physical justification for the conversion of Eq. (9c-f) to the form (10). However, once the procedure is understood, there is no need to interpret it in terms of network modification. One simply identifies the excess elements, formulates the constraints and adds the appropriate terms to the constraint equations. The

[1] It has been shown [1] that when no coupling is present, the augmentation procedure described is minimal in the sense that the number of augmenting elements is the least possible that will eliminate the constraints. When coupling is present other methods are sometimes required to produce a minimal augmentation.

canonical form (2) is thus changed by augmentation to

$$\dot{x} = f(t, x, u) \tag{11a}$$

$$\varepsilon\, \dot{u} = g(t, x, u) \tag{11b}$$

Returning to the example of the multivibrator, we find that the four dependent current sources (VC NR's) are all excess. The left-hand side of the constraint equations (1c-f) associated with these elements must therefore be augmented by the respective terms $\varepsilon\, \dot{u}_3$, $\varepsilon\, \dot{u}_4$, $\varepsilon\, \dot{u}_5$, $\varepsilon\, \dot{u}_6$. Note that these terms correspond physically to the insertion of small capacitances of value ε across each of the excess elements. The resultant equations are therefore almost identical to Eqs. (3) which were derived using the high frequency transistor model. (The only differences are the values assigned to the transistor capacitances.)

The principal justification for the augmentation procedure outlined above is the fact that every physical network element has associated with it some "stray" or "parasitic" energy storage or dissipation which is usually neglected in its mathematical model. Therefore our augmented network model should be a more realistic representation of the true behavior of the physical network than is the constrained model. (More will be said on this subject in Section 6.) The idea of augmentation is, of course, not new: it is commonly used to explain the discontinuous behavior of many types of nonlinear oscillators of the "relaxation" type.

4.3 <u>Numerical Solution of the Augmented Equations</u>. We must now relate some solution of the augmented system (11) to the behavior of the physical system originally described by (2). Since the parameter ϵ was not even present in the original network model, it is reasonable to attempt to minimize the differences between (11) and (2) by considering the limiting behavior of (11) as $\epsilon \to 0+$ as a "true" representation of the physical network. Thus, if $[x(t, \epsilon), u(t, \epsilon)]$ represents a solution of (11) for $\epsilon > 0$, we attempt to determine the limiting solution,

$$[x^o(t), u^o(t)] = \lim_{\epsilon \to 0+} [x(t, \epsilon) u(t, \epsilon)] \qquad (12)$$

At this point we shall assume that a limiting solution (x^o, u^o) exists for which x has a piecewise continuous derivative and u is piecewise continuous. (Although systems of the form (11) have been the subject of considerable study, mainly in the Soviet literature [16, 17], conditions under which the above assumption is valid have not been determined in the general case. For further remarks on this question see Section 6.)

Note that (12) represents, in effect, a solution of an infinitely stiff system. From our previous remarks (Section 2) one might conclude that this would require infinite computation time. As we shall see, however, it is usually easier to integrate an infinitely stiff system than one with finite stiffness.

In computing the limiting solution, we simulate the effect of infinite stiffness as follows:

Let (x_k, u_k) represent the computed values of (x^o, u^o) at the k-th time interval. Noting that in the limit as $\epsilon \to 0+$ the auxiliary variables u move infinitely faster than the state variables x (except when $g = 0$) we shall call (11a) the "slow equations", and replace (11b) at the k-th time interval by the "fast equations"

$$\frac{du}{d\tau} = g(t_{k+1}, x_{k+1}, u) \qquad u(0) = u_k \tag{13}$$

Now, we compute u_{k+1} as

$$u_{k+1} = \lim_{\tau\to\infty} u(\tau)$$

where $u(\tau)$ is the solution of the fast equations (13). That is, u_{k+1} is determined by holding (t, x) constant (i.e., suspending the normal time scale) and integrating the fast equations to steady state. Any convenient numerical integration method can be used on the fast and slow equations, and, of course, some reasonable criterion must be chosen to indicate that the solution of the fast equations has (approximately) reached a steady state. The practicality of the method depends to a large extent on how these tasks are implemented.

5. COMPUTER RESULTS

The algorithm described in Section 4.3 was programmed for an IBM 360/75 computer, using a variable stepsize fourth order Runge Kutta integration scheme for both Eq. (13a) and (13). A simplified flow chart of the program is shown in Fig. 6. Note that each time the program enters the loop for integrating the fast equations it remains there until the norm of the function g is reduced to a sufficiently small value, indicating that a steady state has been (approximately) reached.

To evaluate the efficiency of this computational method, the equations of the multivibrator, Eq. (3), were integrated using the program described in Fig. 6, where Eq. (3e-f) were treated as "fast equations." Note that this simulates the effect of infinitesimally small transistor capacitances. The results were compared with a standard variable step Runge Kutta integration of Eq. (3).

The circuit parameters were as follows: (Units: Volts, mA, kΩ, pF)

$$R_L = 0.6$$

$$R = 6$$

$$E = 10$$

$$C_1 = C_2 = C \text{ (varied)}$$

$$C_C(u) = C_E(u) = 4 \times 10^{-4} e^{40u} + 2$$

$$F_C(V_C, V_E) = 10^{-5}\left[2\left(e^{40V_C} - 1\right) - .98\left(e^{40V_E} - 1\right)\right]$$

$$F_E(V_C, V_E) = 10^{-5}\left[- .98\left(e^{40V_C} - 1\right) + \left(e^{40V_E} - 1\right)\right]$$

In order to observe the effect of the stiffness of Eq. (3) on the efficiency of the computation, the timing capacitor C, and hence the period of the multivibrator, was varied as shown in Table 1. Since the smallest time constant in the circuit is determined by the transistor capacitors C_C, C_E, which were not modified, and since the total solution time is proportional to C*, the "relative stiffness" of the equations is proportional to C as indicated in Table 1.

A computer-generated plot of collector voltage for a typical run (C = 200 pF) is shown in Fig. 7. Since the transistor switching time (based on the high frequency model) is about 6 nanoseconds, the waveform appears to have sharp discontinuities. On a plot of this type, the results of the two different computational methods were essentially indistinguishable.

A reasonable comparison of the relative efficiencies of the two methods can be made on the basis of the total number of individua

* The total solution time was chosen to be about 1.75 times the period of the multivibrator in each case.

function evaluations (NFE) required in each case. (This is a fair measure of required computation time.) Table 1 gives approximate NFE's for each case studied. (One evaluation of the righthand side of one equation is considered to be one function evaluation.) It can be seen from the table that the NFE increases roughly in proportion to the relative stiffness of the system when conventional integration is used. On the other hand, using the method discussed herein, the NFE is effectively independent of the stiffness of the system, resulting in a dramatic increase in efficiency for the larger values of C. These results can be extrapolated to show still more impressive gains in computation time for systems of larger stiffness.

C	50	200	500	
Time interval	750 ns	3000	7500	
Relative stiffness	1	4	10	
NFE (Conventional method)	9.1	26.6	58.7	$\times 10^4$
NFE (Our method)	4.1	3.9	3.8	

Table 1

6. CONCLUSIONS:

We have presented a new computational method of analyzing nonlinear networks on a digital computer. The method was designed to avoid the twin problems of solving constraint equations and/or integrating stiff systems of differential equations. Some of its useful features are:

1) It can be applied whether or not the original network constraint equations are uniquely solvable. When multiple solutions

of the constraint equations exist, the procedure will automatically seek the one which is "correct" on physical grounds. (Subject to certain qualifications stated below.)

2) It is especially useful in computing discontinuous solutions. (Discontinuities or "jump phenomena" often occur in cases where the original network model is indeterminate, and are characteristic of the operation of many types of pulse and digital circuits.)

3) It avoids the numerical problems associated with stiff systems of differential equations.

4) In cases where one is only concerned with gross features of network behavior it offers a convenient means of neglecting "high frequency" effects.

The development of this method is still at a preliminary stage and many open questions remain. One question is that of the existence of the limiting solution (12). For the complex systems to which this method is most suited, there is no general way of predicting the existence and properties of the limiting solutions. A related question, on which almost no work has been done, concerns the validity of the proposed numerical computational scheme in cases where the limiting solution is known to exist. Although empirical results look promising, it would be useful to study the properties of the computational algorithm from a theoretical point of view.

Perhaps the most difficult question, however, pertains to the justification of the method of augmentation on physical grounds. Clearly an infinite variety of augmentations are possible for any network. Stray elements (both lumped and distributed) of any relative values can be inserted throughout the model to account for all of the mechanisms of energy loss and energy storage. In view of the fact that the chosen augmentation is somewhat arbitrary, how can one be sure whether all or only some of the various possible alternative augmentations will accurately reflect the behavior of the physical system under study? In certain simple cases it is possible to prove that all possible augmentations within a given class have essentially the same behavior. [18] However, it seems unlikely that this can be done in the general case. It is, of course, possible that the behavior of the mathematical model will indeed be sensitive to the type of augmentation chosen, and in fact, that limiting solutions may not exist for some types of augmentations. For example, oscillations may occur in (13) leading to the non-existence of a steady state and, perhaps, the non-existence of a limiting solution. The eomputational procedure will fail in this case. However, this is merely an indication of the fact that the physical system being modelled must display the same sensitivity to relative values of its "stray" parameters. Failure of the computational scheme may therefore be taken as a warning of some sort of structural instability of the system under study. Further pursuit of this question would lead us into a general discussion of the philosophy of mathematical modelling. Since this would not be appropriate here, we shall close with the assertion that any uncertainty associated with the validity of the augmented model is no worse than that associated with the original unaugmented model.

ACKNOWLEDGEMENT

It is a pleasure to acknowledge the contribution to this work of Messrs. J. Oso, X. Singhal and S. T. Sathe, who developed a computer algorithm for augmentation, Mr. P. Jacusiel for his variable step Runge Kutta program, and Mr. Y. C. Kim who did most of the computer programming associated with Section 5.

Appendix

Derivation of the Augmented Network Equations

A.1 Partitioning of Variables

Subject to the assumptions of Section 4, a fundamental loop matrix B_f based on any proper tree T will take the form

$$= [I,F] = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|}
C & c & \overbrace{g \; G}^{Y} & z & J & \overbrace{\gamma \; \Gamma}^{\Lambda} & \overbrace{S \; s}^{Q} & V & y & \overbrace{R \; r}^{Z} & \ell & L \\
\hline
I_{CC} & & & & & & F_{CS} & & & & & \\
\hline
 & I_{cc} & & & & & F_{cQ} & & & & & \\
\hline
 & & I_{YY} & & & & F_{YQ} & F_{YV} & F_{Yy} & F_{GR} & & \\
\hline
 & & & I_{zz} & & & F_{zQ} & F_{zV} & F_{zy} & F_{zZ} & & \\
\hline
 & & & & I_{JJ} & & F_{JQ} & F_{JV} & F_{Jy} & F_{JZ} & & \\
\hline
 & & & & & I_{\Lambda\Lambda} & F_{\Lambda Q} & F_{\Lambda V} & F_{\Lambda y} & F_{\Lambda Z} & F_{\Lambda \ell} & F_{\Lambda L} \\
\end{array} \qquad (A-1)$$

where the I's represent identity matrices of appropriate size and the empty partititions of B_f correspond to zero submatrices. (It can be shown that the priority ordering of elements of T always leads to a matrix B_f of this form. The rows and columns of B_f are partitioned into the sets C, c, $\cdots$, L, defined as follows:

$$\left.\begin{array}{l}
\left.\begin{array}{l} C - \text{LPC} \\ \textcircled{c} - \text{Nonlinear} \end{array}\right\}\text{Capacitors} \\
\left.\begin{array}{l} Y\left\{\begin{array}{l} g - \text{VCNR} \\ G - \text{LPR} \end{array}\right. \\ \textcircled{z} - \text{CCNR} \end{array}\right\}\text{Resistors} \\
J - \text{Independent current sources} \\
\left.\begin{array}{l} \Lambda\left\{\begin{array}{l} \gamma - \text{Nonlinear} \\ \Gamma - \text{LPL} \end{array}\right. \end{array}\right\}\text{Inductors}
\end{array}\right\}\text{Chords of } T$$

$$\left.\begin{array}{l}
\left.\begin{array}{l} Q\left\{\begin{array}{l} S - \text{LPC} \\ s - \text{Nonlinear} \end{array}\right. \end{array}\right\}\text{Capacitors} \\
V - \text{Independent voltage sources} \\
\left.\begin{array}{l} \textcircled{y} - \text{VCNR} \\ Z\left\{\begin{array}{l} R - \text{LPR} \\ r - \text{CCNR} \end{array}\right. \end{array}\right\}\text{Resistors} \\
\left.\begin{array}{l} \textcircled{\ell} - \text{Nonlinear} \\ L - \text{LPL} \end{array}\right\}\text{Inductors}
\end{array}\right\}\text{Branches of } T$$

(The circled symbols indicate excess elements)

The fundamental cutset matrix Q_f is derived from B_f:

$$Q_f = [- F^t, I] \qquad (A-2)$$

and Kirchhoff's laws are expressed in terms of B_f and Q_f as

$$B_f \, e_N = 0 \qquad (KVL)$$

$$Q_f \, i_N = 0 \qquad (KCL)$$

where e_N and i_N represent the appropriately ordered branch voltages and currents respectively.

A.2 The Constrained Network Equations

In what follows we indicate variables associated with various subsets of elements by appropriate subscripts. For example, λ_ℓ is a vector (column matrix) whose elements are the flux linkages in each nonlinear tree branch inductor.

Subject to the above method of partitioning we rewrite the element terminal relations given in (6), (7), (8) as follows: (For economy of notation the argument t will be omitted from all functions.)

Capacitive elements:

$$e_S = S\,q_S \quad , \quad q_C = C\,e_C \tag{A-3a}$$

$$e_s = f_s(q_s, q_c), \quad e_c = f_c(q_s, q_c) \tag{A-3b}$$

Inductive elements:

$$i_\Gamma = \Gamma\,\lambda_\Gamma \quad , \quad \lambda_L = L\,i_L \tag{A-4a}$$

$$i_\gamma = f_\gamma(\lambda_\gamma, \lambda_\ell), \quad i_\ell = f_\ell(\lambda_\gamma, \lambda_\ell) \tag{A-4b}$$

Resistive elements:

$$i_G = G\,e_G, \quad e_R = R\,i_R \tag{A-5a}$$

$$\begin{aligned} i_g &= f_g(e_g, e_y, i_r, i_z) \\ i_y &= f_y(e_g, e_y, i_r, i_z) \\ e_r &= f_r(e_g, e_y, i_r, i_z) \\ e_z &= f_z(e_g, e_y, i_r, i_z) \end{aligned} \tag{A-5b}$$

Independent sources:

$$i_J = J, \quad e_V = V \tag{A-6}$$

The state variables $x = (q, \lambda)$, are defined as the cut-set charges,

$$q = q_Q - \begin{bmatrix} F^t_{CS} \\ 0 \end{bmatrix} q_C - F^t_{cQ}\, q_c \qquad \text{(A-7a)}$$

and loop flux linkages

$$\lambda = \lambda_\Lambda + \begin{bmatrix} 0 \\ F_{\Lambda L} \end{bmatrix} \lambda_L + F_{\Lambda \ell}\, \lambda_\ell \qquad \text{(A-7b)}$$

The auxiliary variables are

$$u = (q_c;\ \lambda_\ell;\ e_y;\ i_z) \qquad \text{(A-8)}$$

From the Kirchhoff law relations implied by (A-1) and (A-2) we obtain the following canonical form of the network equations:

$$\dot{q} = f_q(x, u) = F^t_{YQ}\, I_Y + F^t_{zQ}\, i_z + F^t_{\Lambda Q}\, I_\Lambda + F^t_{JQ}\, J \qquad \text{(A-9a)}$$

$$\dot{\lambda} = f_\lambda(x, u) = - F_{\Lambda Z}\, E_Z - F_{\Lambda y}\, e_y - F_{\Lambda Q}\, E_Q - F_{\Lambda V}\, V \qquad \text{(A-9b)}$$

$$0 = g_c(x, u) = - E_c - F_{cQ}\, E_Q \qquad \text{(A-9c)}$$

$$0 = g_\ell(x, u) = - I_\ell + F^t_{\Lambda \ell}\, I_\Lambda \qquad \text{(A-9d)}$$

$$0 = g_y(x, u) = - I_y + F^t_{\Lambda y}\, I_\Lambda + F^t_{zy}\, i_z + F^t_{Yy}\, I_Y + F^t_{Jy}\, J \qquad \text{(A-9e)}$$

$$0 = g_z(x, u) = - E_z - F_{zQ}\, E_Q - F_{zy}\, e_y - F_{zZ}\, E_Z - F_{zV}\, V \qquad \text{(A-9f)}$$

In (A-9) the lower case quantities q, λ, i_z, e_y all represent state or auxiliary variables. The upper case quantities I_Y, E_Z, I_Λ, E_Q, E_c, I_ℓ, I_y, E_z all represent explicit functions of the state and auxiliary variables defined as follows:

Let

$$F_{cQ} = [F_{cS}, F_{cs}], \quad S_N = [S^{-1} + F^t_{CS}\, C\, F_{CS}]^{-1}, \qquad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

(S_N is positive definite, q_1 is of the dimension of q_S, and q_2 of the dimension of q_s.)

Then, combining Kirchhoff's laws with (A-7a) and (A-3) and eliminating the linear terminal relations (A-3a) we obtain,

$$E_Q(q, q_c) = \begin{bmatrix} E_S \\ E_s \end{bmatrix} = \begin{bmatrix} S_N(q_1 + F^t_{cS}\, q_c) \\ f_s(q_2 + F^t_{cs}\, q_c, q_c) \end{bmatrix}$$

$$E_c(q, q_c) = f_c(q_2 + F^t_{cs}\, q_c, q_c) \qquad \text{(A-10)}$$

Let

$$F_{\Lambda \ell} = \begin{bmatrix} F_{\gamma \ell} \\ F_{\Gamma \ell} \end{bmatrix}, \quad \Gamma_N = [\Gamma^{-1} + F_{\Lambda L}\, L\, F^t_{\Lambda L}]^{-1}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

(Γ_N is positive definite, λ_1 is of the dimension of λ_γ and λ_2 of the dimension of λ_Γ.)

Then, combining Kirchhoff's laws with (A-7b) and (A-4) and eliminating the linear terminal relations (A-4a) we obtain

$$I_\Lambda(\lambda, \lambda_\ell) = \begin{bmatrix} I_\gamma \\ I_\Gamma \end{bmatrix} = \begin{bmatrix} f_\gamma(\lambda_1 - F_{\gamma \ell}\, \lambda_\ell, \lambda_\ell) \\ \Gamma_N(\lambda_2 - F_{\Gamma \ell}\, \lambda_\ell) \end{bmatrix}$$

$$I_\ell(\lambda, \lambda_\ell) = f_\ell(\lambda_1 - F_{\gamma \ell}\, \lambda_\ell, \lambda_\ell) \qquad \text{(A-11)}$$

Let

$$F_{YQ} = \begin{bmatrix} F_{gQ} \\ F_{GQ} \end{bmatrix}, \quad F_{Yy} = \begin{bmatrix} F_{gy} \\ F_{Gy} \end{bmatrix}, \quad F_{YV} = \begin{bmatrix} F_{gV} \\ F_{GV} \end{bmatrix}$$

$$F_{\Lambda Z} = [F_{\Lambda R}, F_{\Lambda r}], \quad F_{zZ} = [F_{zR}, F_{zr}], \quad F_{JZ} = [F_{JR}, F_{Jr}]$$

Let

$$E_1 = - F_{gQ} E_Q - F_{gy} e_y - F_{gV} V$$

$$E_2 = - F_{GQ} E_Q - F_{Gy} e_y - F_{GV} V$$

$$I_1 = F^t_{\Lambda R} I_\Lambda + F^t_{zR} i_z + F^t_{JR} J$$

$$I_2 = F^t_{\Lambda r} I_\Lambda + F^t_{zr} i_z + F^t_{Jr} J$$

Let

$$G_N = [G^{-1} + F_{GR} R F^t_{GR}]^{-1} \quad \text{(positive definite)}$$

$$R_N = [R^{-1} + F^t_{GR} G F_{GR}]^{-1} \quad \text{(positive definite)}$$

Then, combining Kirchhoff's laws with (A-5) and eliminating the linear terminal relations (A-5a) we obtain

$$I_y(x, u) = \begin{bmatrix} I_g \\ I_G \end{bmatrix} = \begin{bmatrix} f_g(E_1, e_y, I_2, i_z) \\ G_N(E_2 - F_{GR} R I_1) \end{bmatrix}$$

$$E_Z(x, u) = \begin{bmatrix} E_R \\ E_r \end{bmatrix} = \begin{bmatrix} R_N(I_1 + F^t_{GR} G E_2) \\ f_r(E_1, e_y, I_2, i_z) \end{bmatrix}$$

$$I_y(x, u) = f_y(E_1, e_y, I_2, i_z) \qquad \text{(A-12)}$$

$$E_z(x, u) = f_z(E_1, e_y, I_2, i_z) \qquad \text{(A-13)}$$

A.3 Augmentation

From Fig. 5 we note that when the excess elements are augmented, the expressions (A-10, 11, 12, 13) are modified as follows:

$$E_c \rightarrow \varepsilon\, \dot{q}_c + E_c$$

$$I_\ell \rightarrow \varepsilon\, \dot{\lambda}_\ell + I_\ell$$

$$I_y \rightarrow \varepsilon\, \dot{e}_y + I_y$$

$$E_z \rightarrow \varepsilon\, \dot{i}_z + E_z$$

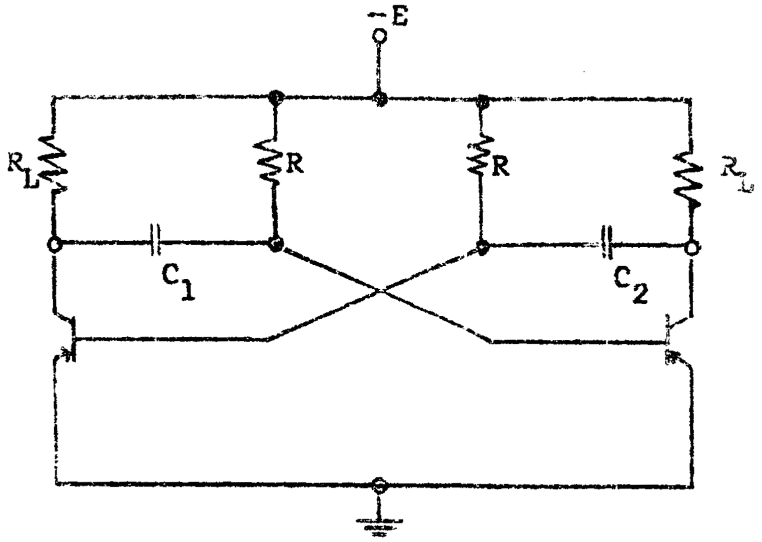
Making these substitutions in (A-9) we obtain the augmented constraint equations:

$$\varepsilon\, \dot{q}_c = g_c(x, u)$$

$$\varepsilon\, \dot{\lambda}_\ell = g_\ell(x, u)$$

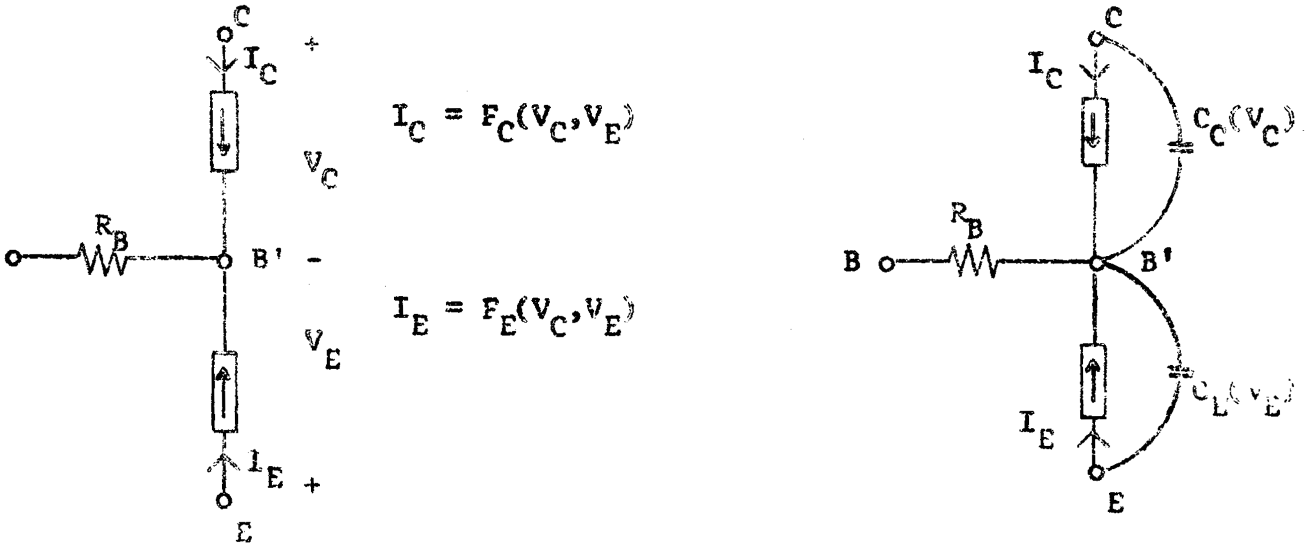
$$\varepsilon\, \dot{e}_y = g_y(x, u)$$

$$\varepsilon\, \dot{i}_z = g_z(x, u)$$



Transistor Multivibrator

Fig. 1



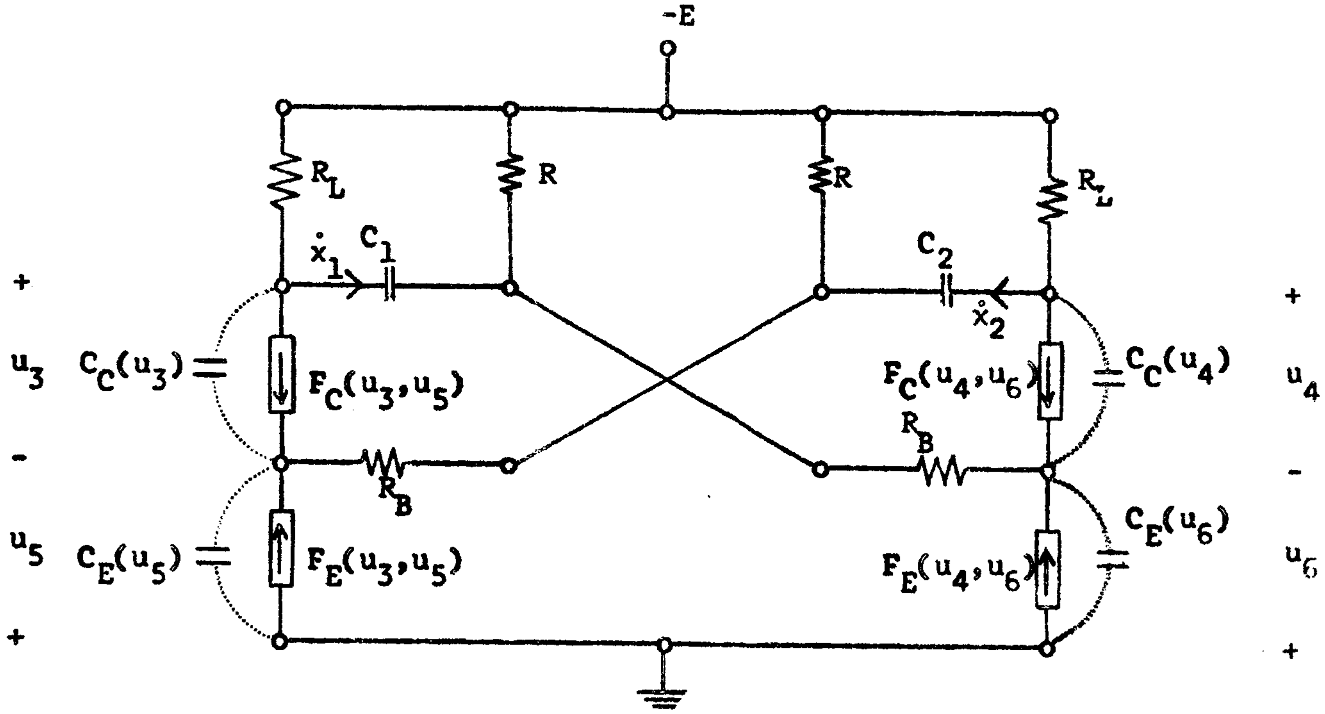
Low frequency transistor model

(a)

High frequency transistor model

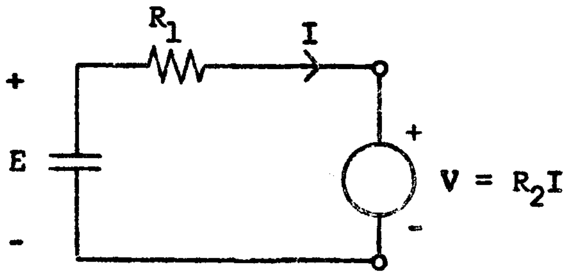
(b)

Fig. 2



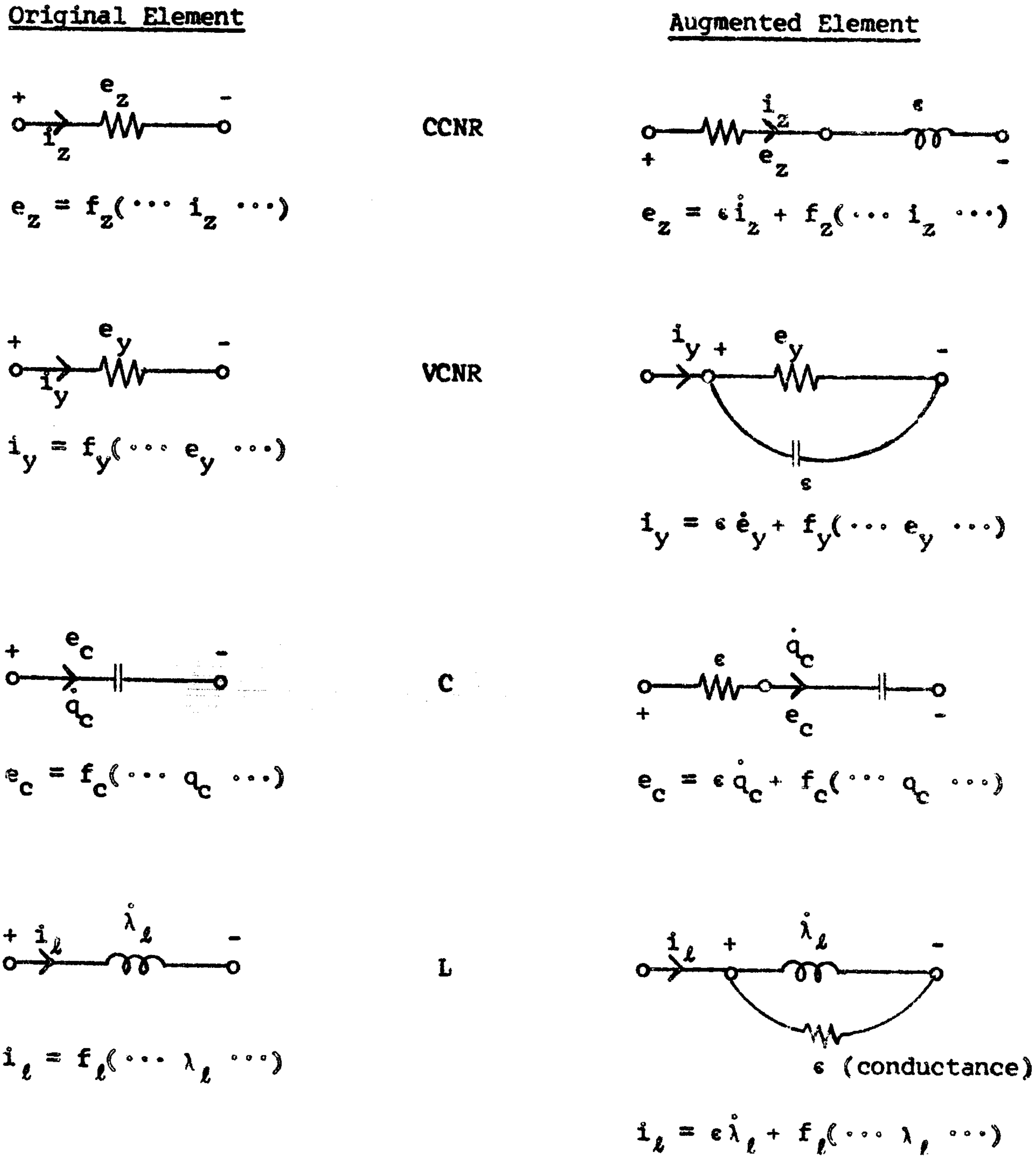
Multivibrator with transistor model

Fig. 3



A "Problem Circuit"

Fig. 4



$$e_z = f_z(\cdots\ i_z\ \cdots)$$

$$e_z = \epsilon \mathring{i}_z + f_z(\cdots\ i_z\ \cdots)$$

$$i_y = f_y(\cdots\ e_y\ \cdots)$$

$$i_y = \epsilon \dot{e}_y + f_y(\cdots\ e_y\ \cdots)$$

$$e_c = f_c(\cdots\ q_c\ \cdots)$$

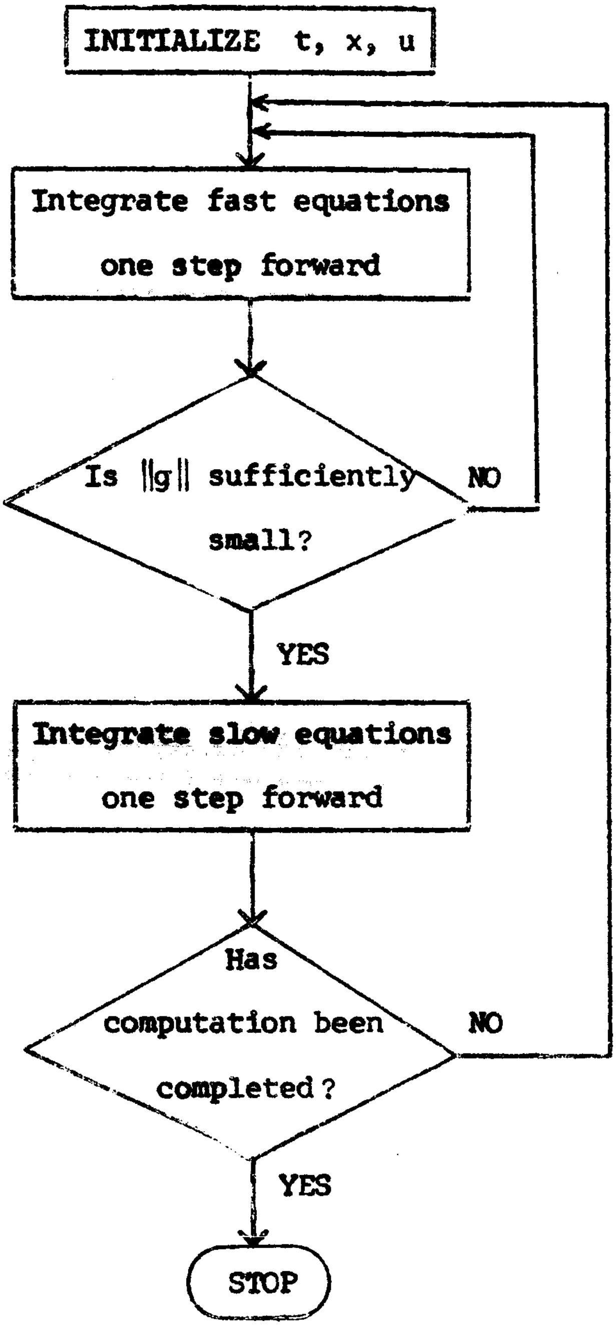
$$e_c = \epsilon \mathring{q}_c + f_c(\cdots\ q_c\ \cdots)$$

$$i_\ell = f_\ell(\cdots\ \lambda_\ell\ \cdots)$$

$$i_\ell = \epsilon \mathring{\lambda}_\ell + f_\ell(\cdots\ \lambda_\ell\ \cdots)$$

Augmentation

Fig. 5

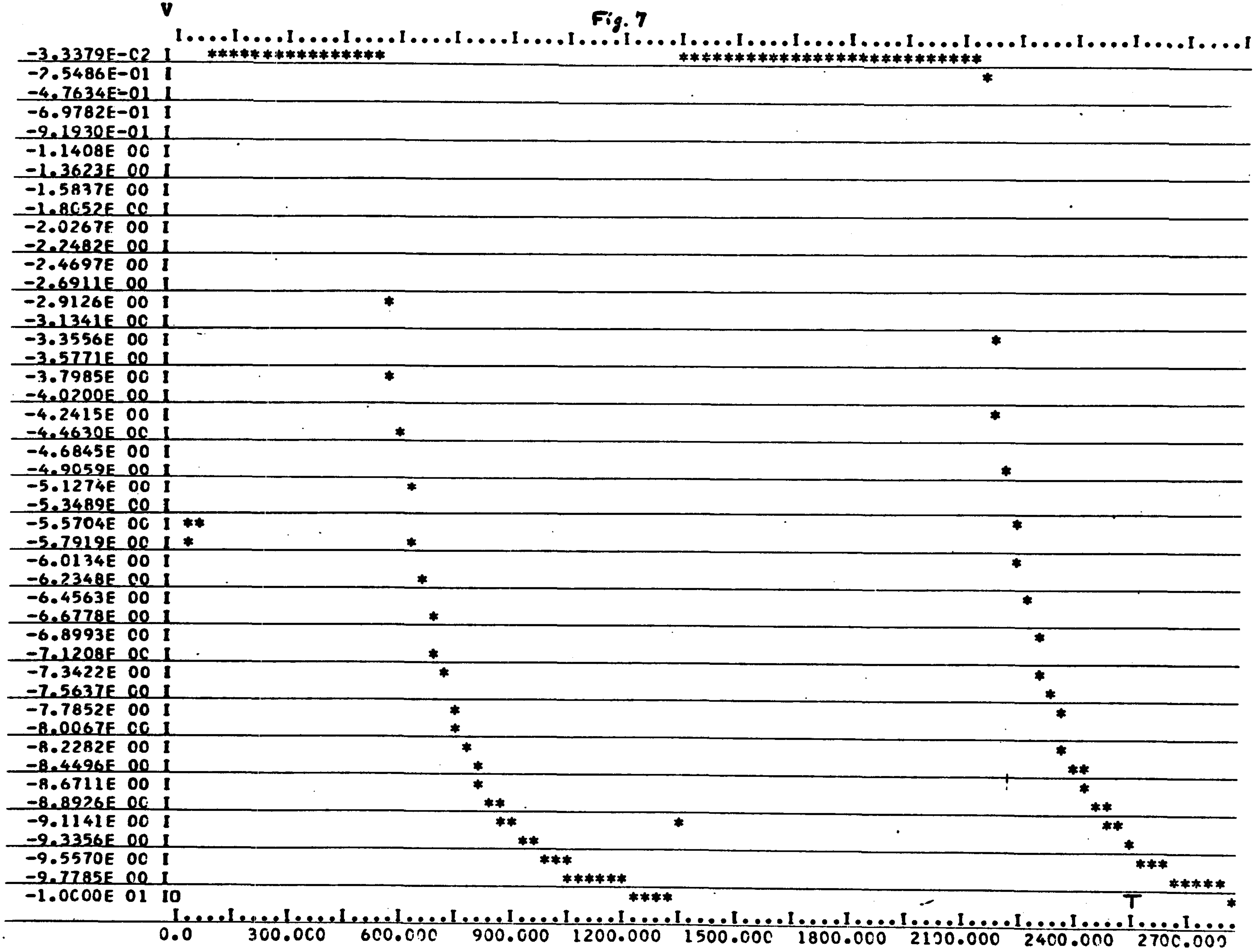


INITIALIZE t, x, u
Integrate fast equations one step forward
Is ||g|| sufficiently small?
NO
YES
Integrate slow equations one step forward
Has computation been completed?
NO
YES
STOP

Flow Chart

Fig. 6

Fig. 7


V
-3.3379E-C2
-2.5486E-01
-4.7634E-01
-6.9782E-01
-9.1930E-01
-1.1408E 00
-1.3623E 00
-1.5837E 00
-1.8C52E CC
-2.0267E 00
-2.2482E 00
-2.4697E 00
-2.6911E 00
-2.9126E 00
-3.1341E 0C
-3.3556E 00
-3.5771E 00
-3.7985E 0C
-4.0200E 00
-4.2415E 00
-4.4630E 0C
-4.6845E 00
-4.9059E 00
-5.1274E 00
-5.3489E C0
-5.5704E 0C
-5.7919E 00
-6.0134E 00
-6.2348E C0
-6.4563E 00
-6.6778E 00
-6.8993E C0
-7.1208E 0C
-7.3422E 00
-7.5637E C0
-7.7852E 00
-8.0067E CC
-8.2282E 00
-8.4496E 00
-8.6711E 00
-8.8926E 0C
-9.1141E 00
-9.3356E 00
-9.5570E 0C
-9.7785E 00
-1.0C00E 01
0.0
300.0C0
600.000
900.000
1200.000
1500.00C
1800.000
2100.000
2400.000
2700.000

REFERENCES

1. S. J. Oh, T. E. Stern and H. E. Meadows, "On the Analysis of Nonlinear Irregular Networks," Proceedings of the Symposium on Generalized Networks, Polytechnic Institute of Brooklyn, April 12, 13, 14, 1966.

2. C. A. Desoer and J. Katzenelson, "Nonlinear RLC Networks," BSTJ 44, no. 1, 161-198 (Jan., 1965).

3. L. O. Chua and R. A. Rohrer, "On the Dynamic Equations of a Class on Nonlinear RLC Networks IEEE Trans. on Circuit Theory CT-12 no. 4, 475-489 (Dec., 1965).

4. T. E. Stern, "On the Equations of Nonlinear Networks," IEEE Trans. on Circuit Theory CT-13 no. 1, (March 1966).

5. W. Liniger and R. A. Willoughby, "Efficient Numerical Integration of Stiff Systems of Ordinary Differential Equations," IBM Research Report RC 1970, (Dec. 20, 1967).

6. P. I. Richards, W. D. Lanning and M. D. Torrey, "Numerical Integration of Large, Highly Damped Nonlinear Systems," SIAM Review 7, 3, 376-380 (1965).

7. C. E. Treanor, "A Method for the Numerical Integration of Coupled First Order Differential Equations with Greatly Different Time Constants," Math. Comp. 20, 39-45 (1966).

8. T. D. Lawson, "Generalized Runge Kutta Processes for Stable Systems with Large Lipschitz Constants," SIAM Journal Num. Anal. 4, 3, 372-380 (1967).

9. M. E. Fowler and R. M. Warten, "A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues," IBM M. Res. Dev. 11, 537-543 (1967).

10. I. W. Sandberg and H. Shichman, "Numerical Integration of Stiff Nonlinear Differential Equations," BSTJ 47, 511-527 (1968).

11. L. D. Milliman, "CIRCUS, A Digital Computer Program for Transient Analysis of Electronic Circuits", Report 346-2, Harry Diamond Labs., Washington, D. C., (Jan. 1967).

12. A. F. Malmberg, F. L. Cornwell and F. N. Hofer, "NET-1 Network Analysis Program," Los Alamos Scientific Lab., Report LA-3119, 7090/94 version, (Aug. 1964).

13. J. Katzenelson, "AEDNET: Simulator for Nonlinear Networks," Proc. IEEE 54 1536-1552, (Nov. 1966).

14. "Automated Digital Computer Program for Determining Responses of Electronic Systems to Transient Nuclear Radiation (PREDICT)", vol. II, IBM Space Guidance Center, Owego, N. Y. IBM File 64-521-5, (July 1964).

15. H. W. Mathers, S. R. Sedore and J. Sents, "Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation, (SCEPTRE)," vol. I, IBM Space Guidance Center, Owego, N. Y., IBM File 66-928-61I, (Feb. 1967).

16. L. S. Pontriagin, "Asymptotic Behavior of the Solutions of Systems of Differential Equations with a Small Parameter in the Higher Derivatives." (Translated from the Russian) AMS Translations Ser. 2 vol. 18, p. 295f. (1961).

17. E. F. Mishchenko, "Asymptotic Calculation of Periodic Solutions of Systems of Differential Equations Containing Small Parameters in the Derivatives" (Translated from the Russian) AMS Translations Ser. 2 vol. 18, p. 187f.

18. S. T. Sathe and T. E. Stern, "The Effect of Stray Parameters on the Stability of a Class of Nonlinear Networks" (to appear).